

## REMARKS

Claims 1, 6-15, 20-28, 37, and 42-49 are pending in this application, with Claims 1, 15, 37, and 49 being the independent claims. Claims 1, 15, 37, and 49 have been amended, and claims 2-5, 16-19, 29-36, and 38-41 are canceled. No new matter is believed to have been added.

### Claim Objections

Independent Claims 1, 15, 37, and 49 were each objected to for including various typographical errors. Applicants regret that the previous amendments to these claims included language that should have been deleted. This amendment corrects these deficiencies, and thus obviates the claim objections.

### Rejections Under 35 U.S.C. § 103

Claims 1, 6, 7, 11, 15, 20, 21, 25, 37, 42, 43, and 47 were rejected under 35 U.S.C. § 103 as allegedly being unpatentable over U.S. Patent Nos. 6,804,664 (Hartman et al.) and 5,710,915 (McElhiney), Claims 8-10, 12-14, 22-24, 26-28, and 44-48 were rejected under 35 U.S.C. § 103 as allegedly being unpatentable over Hartman et al., McElhiney, and U.S. Patent No. 5,201,046 (Goldberg et al.); and Claim 49 was rejected under 35 U.S.C. § 103 as allegedly being unpatentable over Hartman et al., McElhiney, and U.S. Patent Nos. 6,134,500 (Tang et al.). These rejections are respectfully traversed.

Hartman et al. relates to a database that is structured to enable faster, more efficient queries. To do so, the data to be stored in the database is characterized as a number of questions, and each record in the database comprises bit map groups that correspond to the answers to the questions. The answers may be binary attributes, range attributes, and string attributes, depending on the question type. With this type of structure, database queries are obtained by simple bit-wise Boolean operations of the records in the database, beginning first with binary attribute matching, then range attribute matching, and finally string attribute matching. With each attribute matching operation, records in the database are eliminated from the query, thus making the query more efficient (col. 8, l. 9 through col. 12, l. 11).

McElhiney relates to a system and method to easily store and manipulate data in a

relational database system, preferably by implementing what is referred to as “transitive closures” (col. 3, l. 63 through col. 4, l. 42). McElhiney discloses, in the Abstract, a database management system (DBMS) that stores, retrieves, and manipulates directed graph data structures in a relational database. Data are stored in a database in the form of two dimensional tables, which are referred to as flat files. The DBMS defines a schema for each table in the database. The schema defines the name and data type of each column in a database table. Tables that are used to store directed graph data structures include at least one column defined as having a reference data type. Non-empty entries in that column are pointers to rows in a specified table. Directed graph data structures are stored in specified tables by storing each record of the directed graph in a distinct row of one of the specified tables, with references corresponding to interconnections between records being stored in reference data type columns. Portions of a directed graph are retrieved from the specified table, in accordance with a single specified query and then the query is automatically expanded by also retrieving additional portions of the tables which are referenced by the previously retrieved portions, thereby performing a transitive closure. The retrieved data is stored in a buffer as a list of rows, and then communicated to an application process. An interface program converts the list of rows stored in the buffer into a directed graph data structure.

The Office action alleges that Hartman et al. discloses a database that is compatible with multiple end-user systems. The Office action further alleges that the content database of Hartman et al. corresponds, at least generally, to the data section of independent Claims 1, 15, 37, and 49 of the instant application, and cites col. 4, ll. 35-46, col. 6, ll. 19-24, and col. 7, ll. 39-48 to support this allegation. The Office action also alleges that Hartman et al. discloses, at least generally, the structure section of independent Claims 1, 15, 37, and 49 of the instant application, and cites col. 6, ll. 25-38, col. 7, ll. 16-26, and col. 8, ll. 9-18 and 54-64 to support this allegation. The Office action then goes on to opine that the only deficiency of Hartman et al. is that it does not disclose that the data section and the structure section each comprise a plurality of tables.

To make up for the lacunae of Hartman et al., the Office action cites McElhiney. Specifically, the Office action cites col. 7, ll. 49-58 of McElhiney as disclosing “the partitioning of a data table into a plurality of tables.” The Office action then goes on to

conclude that it would have been obvious “to partition the data section and the structure sections of Hartman into a plurality of tables . . . to provide parallel access to the tables which accelerates access.” Office action at 4-5. As will now be explained, the analysis proffered in the Office action is inaccurate, and furthermore the combination of Hartman et al. and McElhiney does not establish a prima facie case of obviousness.

First of all, Hartman et al. does not disclose all that is alleged in the Office action. Specifically, and as was noted above, the Office action alleges that that the content databases (150, 151, 152) of Hartman et al. corresponds to the data section. The Office action references col. 4, ll. 35-46, col. 6, ll. 19-24, and col. 7, ll. 39-48 to support this allegation. However, all that col. 4, ll. 35-46 states is:

The database of the invention is built from records, profiles, targets and attributes. An “attribute” may be of the following types: binary attribute, range attribute or string attribute. As described further below, an attribute’s type defines the type of value which that attribute may have. Other attribute types are within the scope of the invention. Attributes are similar to database fields. A “target” consists of one attribute and a value for that attribute. A “profile” is made up of targets, or more particularly, attribute values. A “record” is an entity having the attributes of one or more profiles. A record is one unit of storage in a database, though records may be comprised of profiles in multiple databases.

Column 6, ll. 19-24 states:

The correlation table 160 is used in conjunction with binary attributes. The correlation table 160 shows the connection between each answer and the bit that represents it. The correlation table 160 shows how to encode binary attributes, providing static mapping between allowable value and bit location.

And col. 7, ll. 39-48 states:

For those fields of the raw data which correspond to binary attributes, the database server 130 looks up the fields in the correlation table 160 and retrieves the bitmask for the binary attributes corresponding to the fields (step 220). For range attributes and string attributes, some processing may also be useful or necessary prior to storing. For example, range values may be rounded, and string values may be concatenated and parsed for illegal characters. The retrieved bitmask, plus the values of the range attributes and string attributes, form one or more profiles of the record.

Interestingly, none of the above portions of Hartman et al. even remotely address the content databases (150, 151, 152). More significantly, however, according to the

independent claims of the instant application: (1) each data table that comprises the data section includes a plurality of *data records that each have one or more features that affect its compatibility with one or more of the end-user systems*, and (2) *each data record includes a feature field that contains one or more feature bits representative of each of its features*. Although Hartman et al. discloses looking up fields in the correlation table and retrieving bitmasks for the attributes corresponding to the fields, nowhere does Hartman et al. disclose, in those portions referenced in the Office action or any other portion, data records having one or more features that affect compatibility with one or more of end-user systems or data records that include a feature field that contains one or more feature bits representative of each of its features.

The Office action also alleges that Hartman et al. discloses a structure section. The Office action references col. 6, ll. 25-38, col. 7, ll. 16-26, and col. 8, ll. 9-18 and 54-64 to support this allegation. However, all that col. 6, ll. 28-38 states is:

The user profile database 140 stores and provides information about users. In some cases, such as with an Internet service provider, the users may provide some user profile information directly. User profile information may be obtained from existing databases or through indirect means (e.g., analysis of user habits, purchasing histories). Users may be required to or may voluntarily update their user profiles on a periodic basis. The user profiles may include demographic information such as age, sex, marital status, home address and hobbies, as well as psychographic information such as likes and dislikes. Information about the client devices used by each user, such as type of device and processing capabilities, may also be obtained and stored in the user profile database 140 or another database.

Column 7, ll. 16-26 states:

The client profile database 170 stores and provides profiles of client devices 100. Client profiles may include such information a software versions, processor type, processor speed, memory size, modem type, etc. The client profile database 170 is related to the user profile database 140, in that the profiles of client devices 100 used by the users are stored in the client profile database 170. User profiles may refer to client profiles, and client profiles may refer to user profiles. This is because a given client may be used by multiple users, and a given user may use multiple client devices.

And col. 8, ll. 9-18 and 54-64 states:

As described below, the method of querying a database is a reductive process. In a typical database query, the fields of each record of the database is

compared in turn against the query. If the fields of a record match the query, then the record is added to a list of matching records. A typical database query is therefore an additive process. According to the invention, a query is structured to benefit from how the data is stored (as described above), and how a query may be best processed. The present invention may also be considered a successive filtering process.

According to the independent claims: (1) each feature mask table that comprises the structure section includes a feature mask record for each of the multiple end-user systems that use one or more of the data tables that include the data records having one or more features, and (2) *each feature mask record includes one or more feature mask values that indicate whether the one or more features of a data record are compatible with one or more of the end-user systems, and thereby indicate whether the data record is compatible with one or more of the end-user systems.* Although Hartman et al. discloses storing user profile data and client profile data, and that user profiles may refer to client profiles, and vice-versa, the reason is because multiple users may use a particular client, and a particular user may use multiple clients. This has nothing whatsoever to do with data records having features that affect the data record's compatibility with an end-user system. Rather, this deals with whether a user, via a specific client, may access certain data. As has been stated repeatedly during the prosecution of this application, accessibility to data by a device or system, and compatibility of data with a device or system, are completely different issues. It is the former at which, at best, Hartman et al. even hints.

As regards McElhiney, it was cited for allegedly "partitioning of a data table into a plurality of tables." Office action at 4. However, all that this reference discloses is storing data in a database in the form of two dimensional tables, and not partitioning a data table into many data tables. Nonetheless, even if one were to concede that McElhiney discloses what the Office action alleges, it does not make up for the rather glaring deficiencies of Hartman et al. with respect to the independent claims. Therefore, the combination of Hartman et al. and McElhiney does not, indeed cannot, establish a *prima facie* case of obviousness.

In view of the foregoing, reconsideration and withdrawal of the § 103 rejections is requested.

### Conclusion

Based on the above, independent Claims 1, 15, 37, and 49 are patentable over the citations of record. The dependent claims are also deemed patentable for the reasons given above with respect to the independent claims and because each recite features which are patentable in its own right. Individual consideration of the dependent claims is respectfully solicited.

The other art of record is also not understood to disclose or suggest the inventive concept of the present invention as defined by the claims.

Hence, Applicant submits that the present application is in condition for allowance. Favorable reconsideration and withdrawal of the objections and rejections set forth in the above-noted Office action, and an early Notice of Allowance are requested.

If the Examiner has any comments or suggestions that could place this application in even better form, the Examiner is requested to telephone the undersigned attorney at the below-listed number.

If for some reason Applicant has not paid a sufficient fee for this response, please consider this as authorization to charge Ingrassia, Fisher & Lorenz, Deposit Account No. 50-2091 for any fee which may be due.

Respectfully submitted,

INGRASSIA FISHER & LORENZ

Dated: July 7, 2009

By: /PAUL D. AMROZOWICZ/  
Paul D. Amrozowicz  
Reg. No. 45,264  
(480) 385-5060